

Docket No: POU920000073US1

A METHOD AND SYSTEM OF
AUTOMATIC DELAY DETECTION AND
RECEIVER ADJUSTMENT FOR
SYNCHRONOUS BUS INTERFACE

APPLICATION FOR
UNITED STATES LETTERS PATENT

Express Mail Label No: EJ568510221US

Date of Deposit: April 2, 2001

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee Service under 37 CFR 1.10 on the date indicated above and is addressed to Box Patent Application, Commissioner of Patents and Trademarks, Washington, D. C. 20231.

Susan L. Nelson 

I N T E R N A T I O N A L B U S I N E S S M A C H I N E S
C O R P O R A T I O N

A METHOD AND SYSTEM OF AUTOMATIC DELAY DETECTION AND RECEIVER ADJUSTMENT FOR SYNCHRONOUS BUS INTERFACE

FIELD OF THE INVENTION

[0001] The present invention relates generally to synchronous bus transactions in digital systems, and more particularly to such systems in which the delays of the interface are greater than the system cycle time.

BACKGROUND OF THE INVENTION

[0002] In a synchronous digital system, especially in a computer system, synchronous buses are preferred and commonly used in between subsystems as interface buses (usually chip-to-chip interfaces) to achieve the high bandwidths required for those systems. For synchronous bus transactions to function properly, both sender (multiple bits of binary lines) and receiver sides must operate on a synchronous clock from a common clock oscillator and possibly via phase locked loop circuits on both sides for high frequency systems. Another requirement is that the receiver side must capture all the signal bits of the bus on the same clock cycle to maintain the synchronization.

[0003] When the worst-case delay from the sender flip-flops or latches to the receiver flip-flops (FF) (minus the FF setup time) is less than one cycle time (clock period) and the best-case delay is more than the FF hold time, the synchronous bus transaction across the interface has one-cycle delay. That is the simplest synchronous bus interface. However, in high frequency synchronous systems wherein the worst-case delay is more than one cycle and the delay differences among the bits/lines of the bus are more than the cycle time, synchronous bus transactions become much more difficult.

[0004] This synchronous bus transaction delay problem has been addressed by others by mainly two approaches. The first approach is to fine tune all the bits of a synchronous bus by adding extra delay (extended wire length) to the bit lines with less delay. This approach tightens the interface physical design requirement and complicates the system design; it is still difficult to determine in which cycle to capture data even with this approach at high frequencies.

[0005] The second known approach is using various source-synchronous techniques to align signal bits and the clock (such as "Dynamic Wave-pipelined Interface Apparatus and Methods Therefor." October 1999, IBM Patent Docket- AT998212), as well as signal buffering and rotations at the receiver side (such as "An Elastic Interface Apparatus and Method Therefor." October 1999, IBM Patent Docket- AT998208). Some of those source-synchronous interface and buffering techniques are very sophisticated, but these techniques are also complicated and their implementations require larger circuitry than the method and system of this invention.

SUMMARY OF THE INVENTION

[0006] The present invention is a method of automatic delay detection and receiver adjustment for a synchronous bus with multiple-cycle delays. The primary application of this invention is signal and data transfers between integrated-circuit chips in a synchronous system, but this invention's applications are not limited to chip interfaces alone.

[0007] A system implemented with this method is capable of determining whether the sender to receiver delay of every signal (or data) bit is more than the projected capture cycle time (not necessarily one cycle) during a detection and calibration period and to

store this delay information in the receiver circuitry. Then the receivers are individually adjusted to receive signals at the appropriate cycle time based on the delay information. Therefore, a synchronous interface implemented with this invention will be able to transfer data with a minimum number of cycles across the interface that is optimized based on the delay limit of that particular bus interface thus achieving the best bandwidth possible.

[0008] Because the receiving time of each receiver (capture time of each receiver flip-flop or latch) is adjusted based on its sender to receiver delay, it is not required that all data bits must arrive at the receivers within the one cycle prior to the capture clock edge of the receiver flip-flops that transfer the signals to the receiving chip internal logic.

[0009] This method of automatic delay detection and receiver adjustment for synchronous bus transactions allows higher frequency transfer and achieves much better tolerance to bus bit misalignments than conventional synchronous bus transactions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The objects and features of the invention will be understood by a reading of a detailed description, taken in conjunction with the drawings, in which:

[0011] FIG. 1 sets forth a block diagram of a typical synchronous bus interface;

[0012] FIG. 2 illustrates circuitry for delay detection in the sender side of a bus interface arranged in accordance with the principles of the invention;

[0013] FIG. 3 illustrates circuitry for delay detection, calibration, and normal operation in the receiver side of the bus interface;

[0014] FIG. 4 illustrates an implementation of the Worst Delay Detect Logic of the system for one receiver bus bit in accordance with the invention; and

[0015] FIG. 5 illustrates the logic timing of the system of FIGS. 2 and 3 when the worst-case delay is less than 1.5 clock periods.

DETAILED DESCRIPTION OF THE INVENTION

[0016] In a digital system with synchronous bus interface 100 as shown by FIG. 1, the sender subsystem 102 includes one or more integrated circuit chips, such as Chip A, and the receiver subsystem 104, including one or more chips, such as Chip B, maintain clock synchronization by a common clock reference 106. This common clock source 106 is usually provided by a clock/crystal oscillator circuitry or chip. Both subsystems 102 and 104 minimize clock skew through some clock distribution techniques and in most cases with phase-locked loops (PLL) 108A and 108B in high frequency systems. FIG. 1 shows Chip A 102 as the sender and Chip B 104 as the receiver. However, in a bi-directional bus interface, a subsystem, such as Chip A or Chip B in FIG. 1, is a sender and a receiver. FIG. 1 illustrates a unidirectional N-bit bus interface in which the sender flip-flops (FF) 104-I through 104-N, or master-slave latch pairs in a level-sensitive scan design (LSSD), in Chip A 102 transfer signals to the receiver flip-flops (FF) 106-I through 106-N or latches in Chip B 104.

[0017] To determine the sender to receiver delay, a system arranged in accordance with the principles of this invention first goes through a delay detection and calibration phase. During this phase, the sender 202 as shown by FIG. 2 launches a stream of test patterns for a few thousand cycles. The simplest test pattern is a “10101...” toggle pattern generated by the Calibration Pattern Generator logic 204 in FIG. 2. A variety of other test patterns, such as an “110110...” pattern, can be used to improve the accuracy and reliability of delay detection and calibration for the system. The *set(synch.)* 206 signal to the

FF 208 of the Calibration Pattern Generator 204 is to illustrate that the test patterns need to be synchronized with the *select* signal 350 of the Worst Delay Detect Logic 306 in FIG. 3. The test patterns are used by the receiver side 304 (FIG. 3) to determine the worst-case (longest) delays in the few thousand test cycles.

[0018] FIG. 3 illustrates the receiver circuits with the Worst Delay Detect Logic circuits 306 of this invention. During the delay detection and calibration phase, the output signal *wc_ok* 308 of the Worst Delay Detect Logic for every bit of the bus is determined. If the delay of the test pattern received in a receiver bit is ever more than the targeted capture clock edge during the delay detection phase, signal *wc_ok* 308 for that bit is cleared as a logic 0. Otherwise, it stays as logic 1. The output signals (*wc_ok*'s) from all the Worst Delay Detect Logic circuits 350 are used by the Clock & Signal Select Control Logic 310 as shown in FIG. 3 to select the MUX's 312-I through 312-N and the clock for FF2's 314-I through 314-N of the receivers.

[0019] In case all *wc_ok*'s are at logic 1's, the NAND gate 316 output 318 of the Clock & Signal Select Control Logic 310 in FIG. 3 is at logic 0. The incoming bus signals Bit[0:N] instead of the output of FF1's 320-I through 320-N are selected as the input of FF2's 314-I through 314-N. The same clock used for Worst Delay Detect Logic 306-I thorough 306-N is selected as the clock for FF2's to capture the bus signals Bit[0:N].

[0020] If one or more output signals of the Worst Delay Detect Logic 306 is at logic 0, the output of the NAND gate 316 in FIG. 3 is set to logic 1. The bus bit whose Worst Delay Detect Logic output *wc_ok* 308 is at logic 1 will select FF1 output as FF2 input to capture the early bit at an early time. The bus bit whose Worst Delay Detect Logic output *wc_ok* 308 is at logic 0 will select the bus bit directly as FF2 input to capture the late bit at a

later time. The inverted clock 341 is selected as the clock for FF2's as well as for the internal logic 360 of the chip 304. Then the system is ready for normal operation.

[0021] The delay detection and calibration phase of the system must be done first before the system is ready for the bus transfers. Normally this phase is carried out within a few thousand cycles when the system is first powered up. During the system normal operation, it is not necessary to run this delay detection and calibration procedure again.

[0022] It is necessary to emphasize here that having the sender of chip 202 and receiver of chip 304 subsystems operate with 180° out-phased clocks (inverted clocks for FF2 of Chip B 360 and the internal logic in FIG. 3 when one or more output signals of the Worst Delay Detect Logic 306 is at logic 0) is one typical implementation. In a system with multiple subsystems wherein it is inconvenient or difficult to have out-phased clocks for the chip internal logic blocks among senders and receivers, the implementation of this invention requires to have the sender FF's 210-I through 210-N in FIG. 2 and the receiver FF1's 320-I through 320-N in FIG. 3 switch to the 180° out-phased clock in reference to FF2's 314-I through 314-N in FIG. 3 and the sender and receiver internal logic 212 and 360 when one or more output signals of the Worst Delay Detect Logic is at logic 0. In such an alternative implementation, a 1-bit indicator signal from the receiver to the sender or through other communication method between them is needed for the sender to select non-inverting 340 or inverting 341 clock for its driver FF's 314-I through 314-N. It is also necessary to make sure that the combination logic paths from the sender internal logic 212 to the driver FF's do not violate the driver FF setup time when the sender FF's operate on the 180° out-phased clock. There are many known methods to address such timing issues.

[0023] The most common application of this system is for worst-case delay less than 1.5 clock periods. The skew among all the bus bits can be almost 1.5 clock periods instead of less than 1 clock period required by a conventional synchronous bus interface. The requirement is that the best-case delay must be within 1 cycle prior to the clock capture edge of the Worst Delay Detect Logic and the worst-case delay cannot be more than 0.5 cycle later than the same clock edge. Another requirement for the system of this invention is that the best and worst case delay of a bus bit for any bit patterns in the specified environment (temperature, voltage, etc.) must be less than 0.5 of the clock period.

[0024] For a bus interface that the best and worst-case delay is between $\frac{1}{2}$ and 2 clock periods, the clock for the Worst Delay Detect Logic needs to be out-of-phase in reference to the sender chip clock. This invention can also be used in cases where the worst-case delay is more than 2 clock periods/cycles as long as the bus meets the timing requirements described in the previous paragraph. The clock frequencies of the sender or receiver subsystem can also be the multiples or a half of the frequency of the bus transaction in which cases the system requires minor modifications in the implementations.

[0025] One variation of this method and system is to send the driver clock with the bus bits as the clock for the receiver Worst Delay Detect Logic and FF1's 320-I through 320-N in FIG. 3. The advantage of doing so is that FF1's 320-I through 320-N are source-synchronized so there are less PLL jitter problems at high frequencies. The received driver clocks must have a known phase relationship with the receiver clock.

[0026] It is also possible to have multiple Worst Delay Detect Logic circuits operate at different clock phases for every bus bit for more precise delay detection and finer steps of receiver FF capture times.

[0027] FIG. 4 illustrates a typical implementation of the Worst Delay Detect Logic 306 of FIG. 3. An optional delay element 402 is added to delay the bus bit in a fraction of the clock period to compensate the worst-case delay difference between the “10101...” test pattern and a real signal which is usually worse than the test pattern. The *set* signal 406 to FF4 408 (scan initialize is an alternative) in FIG. 4 is used to set the output *wc_ok* 308 to a logic 1 prior to the delay detection and calibration phase. As FIG. 5 shows, the *select* signal 410 of FIG. 4 is a toggle pattern which is synchronized with the “10101...” test pattern so that FF3 412 samples the test pattern in alternating cycles for delay detection. The *calibrate* signal 414 in FIG. 4 can only be enabled (to logic 1) when the receiver input is the “10101...” test pattern. As FIG. 5 shows, the Worst Delay Detect Logic output *wc_ok[0]* stays at logic 1 for Bit[0] whose worst-case arrival time is earlier than the clock rising edge, and the Worst Delay Detect Logic output *wc_ok[N]* switched to logic 0 for Bit[N] whose worst-case arrival time is later than the clock rising edge during the delay detection phase when the *calibrate* signal 414 in FIG. 4 is at logic 1.

[0028] As shown in FIG. 5, the sender Chip A 202 sends the “10101...” test pattern in all bus bits during the delay detection phase. The dashed lines 501 for Bit[0] and 502 for Bit[N] wave forms at the receiver Chip B 304 indicate the earliest possible bit arrival times and the solid lines 501A, 502A indicate the latest possible bit arrival times. FIG. 5 shows the case that the worst-case delay among all bus bits is less than 1.5 of the clock period.

[0029] After the calibration in the delay detection phase, the early-arrival Bit[0] is captured by FF1 320-I of FIG. 3 (shown as Bit[0].FF1.Q as FF1 output) then transferred to FF2 314-I to synchronize with the late-arrival bus Bit[N]. The bus bits can be any pattern instead of the toggle pattern shown in FIG. 5.

[0030] While the preferred embodiment to the invention has been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first described.